Journal of Information Systems and Telecommunication
Vol.10, No.2, April-June 2022, 111-119
http://jist.acecr.org
ISSN 2322-1437 / ESSN:2345-2773

Optimized kernel Nonparametric Weighted Feature Extraction for Hyperspectral Image Classification

Mohammad Hasheminejad[1*]

[1].Department of Electrical Engineering, University of Jiroft, Kerman, Iran

Received: 13 May 2021/ Revised: 04 Oct 2021/ Accepted: 21 Dec 2021
DOI:

Abstract

Hyperspectral image (HSI) classification is an essential means of the analysis of remotely sensed images. Remote sensing of natural resources, astronomy, medicine, agriculture, food health, and many other applications are examples of possible applications of this technique. Since hyperspectral images contain redundant measurements, it is crucial to identify a subset of efficient features for modeling the classes. Kernel-based methods are widely used in this field. In this paper, we introduce a new kernel-based method that defines Hyperplane more optimally than previous methods. The presence of noise data in many kernel-based HSI classification methods causes changes in boundary samples and, as a result, incorrect class hyperplane training. We propose the optimized kernel non-parametric weighted feature extraction for hyperspectral image classification. KNWFE is a kernel-based feature extraction method, which has promising results in classifying remotely-sensed image data. However, it does not take the closeness or distance of the data to the target classes. Solving the problem, we propose optimized KNWFE, which results in better classification performance. Our extensive experiments show that the proposed method improves the accuracy of HSI classification and is superior to the state-of-the-art HIS classifiers.

Keywords: Feature Extraction; Image Classification; Optimized KNWFE; Hyperspectral; Kernel.

1- Introduction

Hyperspectral image (HSI) classification is widely used in many fields such as agriculture, mineralogy [1], environmental monitoring, and material analysis [2]. An HSI image contains spatial-spectral information, which is the visible and near-infrared, and short-wavelength infrared spectrum, for different locations in an image plane. This image plane is usually obtained by airborne and spaceborne spectrometers [3]. These images have many spectral bands and complex spatial structures containing lots of information. These images typically cover a wide spectral range of frequencies. As a result, each pixel vector is a highly-detailed spectral representative of each captured land cover material. Therefore, since the types of materials on the ground are better identified using HSI images, they can be used in many applications performed via surface analysis. The analysis of HSI involves classification. The goal of classification is to assign a unique class label to each pixel vector.

As an example of HSI classification methods, SVM can be cited [4]. SVM searches an optimal hyperplane to separate the data in a multi-dimensional feature space. Other widely used spectral classification methods include k-nearest-neighbors, maximum likelihood, logistic regression, neural networks [5]. To avoid the computational burden and increase the classification accuracy, it is recommended to use dimensionality reduction techniques [6]. In the past several years, many feature extraction and classification methods have been presented for hyperspectral data [1], [7]. An example of supervised dimensionality reduction is linear discriminant analysis[8]. Besides, non-parametric weighted feature extraction (NWFE) [9], local joint subspace (LJS) detection [4], independent component analysis [10], principal component analysis [11], superpixelwise PCA [12], and semi-supervised discriminant analysis (SDA) [12] are dimensionality reduction methods which are considered by the community. However, due to the unbalance between the limited number of training samples and the high dimensionality of data, HSI classification is still a highly challenging task [13].

In hyperspectral image classification, each pixel is labeled with one of the classes based on its features. SVM is known as a powerful method in HSI classification [14]. Another classifier that is widely used is multinomial

✉ **Corresponding Author**
mhn@ujiroft.ac.ir

logistic regression [15]. This classifier uses the logistic function to provide the posterior probability. In [15], an ensemble multinomial logistic regression-based method is used for HSI classification. An anomalous component extraction framework for the detection of hyperspectral anomalies based on Independent Component Analysis (ICA) and orthogonal subspace imaging (OSP) is proposed in [16]. Kernel-based SVM approaches can offer satisfying performance in HSI classification. Mountrakis et al. showed that using a nonlinear kernel with a local k-nearest neighbor adaptation improves the performance of localized types of SVM approaches [17]. A regularization method is proposed in [18] to address the issue of kernel predetermination. The technique identifies kernel structure through the analysis of unlabeled samples. H. C. Lee et al. proposed an HSI classifier that projects Gabor features of the hyperspectral image into the kernel induced space through composite kernel technique [1]. Representation-based methods such as sparse representation are proven to be promising in pattern recognition. HSI sparse representation classification is based on the assumption that pixels belonging to the same class lie in the same subspace. It is also applied to HSI classification [19], where the representation is performed in a feature space induced by a kernel function. Sparse representation classification is now a popular method in hyperspectral unmixing. Weng et al. used a kernel to map hyperspectral data and library atoms to a suitable space to unmix hyperspectral information [20]. Sparse representation is also used to enhance hyperspectral images [21].

Recently, a variety of deep learning-based algorithms has shown their promising performance in various applications, including HSI classification [22]. Due to the success of deep learning in the field of pattern recognition, it has attracted many researchers in hyperspectral image classification and analysis [23], [24]. In [23], a convolutional neural network (CNN) architecture is proposed for HSI classification. They proposed a 3-D network that uses both spectral and spatial information. To effectively process the border areas in the image, it implemented a border mirroring strategy. The proposed algorithm is implemented on graphical processing units. In [24], a simplified deep neural network is proposed. This network, which is called MugNet, utilizes the relationship between different spectral bands and neighboring pixels. It also generates a convolution kernel using a semi-supervised manner. The application of deep SVM in HSI classification is investigated in [25]. Four kernel functions were used in that study.

However, it is commonly necessary to pre-process that spectral information to use in HSI analysis. This process includes reducing the number of bands using proper techniques. In this case, non-parametric weighted feature extraction (NWFE) has shown promising results in HSI dimension reduction [9]. It is further improved in [26] as KNWFE, taking advantage of the kernel method. In this paper, we try to improve within and between class scattering matrices, correcting data weightings.

The rest of this paper is organized as follows: Section 2 overviews the KNWFE method. In Section 3, we propose our corrections on the KNWFE followed by the performed experiments in Section 4. We conclude in Section 5.

2- Related Work

Most of the time, HSIs are not linearly separable. Therefore kernel methods are used to project the data into a feature space, where the classes are linearly separable. The kernel function is a similarity function that corresponds to an inner product in some expanded feature space. Some popular kernel functions are linear kernel, polynomial kernel and gaussian radial-basis-function (RBF) kernel.

The proposed algorithm is a nonlinear kernel-mode based on the nonparametric weighted feature extraction (NWFE) method [26]. NWFE is a nonparametric method for high-dimensional multi-class pattern recognition problems. This algorithm is based on a non-parametric expression of the scatter matrix. The steps of this algorithm are to first calculate the Euclidean distance between each sample pair and place it in a matrix called the distance matrix. Then the weights matrix is calculated using the distance matrix. The weighted mean matrix is then calculated by putting different weights on every sample. Then, the distance between samples and their weighted means is calculated, as their closeness to the boundary. Finally, nonparametric between-class and within-class scatter matrices are defined, to put large weights on the samples close to the boundary and deemphasize samples far from the boundary. These matrices are defined respectively as [26]:

$$S_b^{NW} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right)^T$$

$$S_w^{NW} = \sum_{i=1}^{L} P_i \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right)^T$$

where $\lambda_\ell^{(i,j)}$ is scatter matrix weight and is defined by:

$$\lambda_\ell^{(i,j)} = \frac{dist\left(x_\ell^{(i)}, M_j\left(x_\ell^{(i)}\right)\right)^{-1}}{\sum_{t=1}^{N_i} dist\left(x_t^{(i)}, M_j\left(x_t^{(i)}\right)\right)^{-1}}$$

with $M_j(x_\ell^{(i)}) = \sum_{k=1}^{N_j} \omega_{\ell k}^{(i,j)} x_k^{(j)}$, that denoted the weighted mean concerning $x_\ell^{(i)}$ in class j, $dist(A,B)$ the distance between A and B, and

$$\omega_{\ell k}^{(i,j)} = \frac{dist\left(x_\ell^{(i)}, x_k^{(j)}\right)^{-1}}{\sum_{t=1}^{N_j} dist\left(x_\ell^{(i)}, x_t^{(j)}\right)^{-1}}$$

Despite that NWFE has better performance than LDA, it is still linear. The KNWFE method, a kernel-based nonlinear version of the NWFE, is presented to derive the non-Gaussian data feature [26]. In this method, $x_\ell^{(i)}$ in the scatter matrices is replaced by $\varphi(x_\ell^{(i)})$, where $\varphi(.)$ is a kernel function.

2-1- Kernel Nonparametric Weighted Feature Extraction

The strategy of kernel-based methods is to map data from the original space to a higher-dimensional Hilbert space, where the data are expected to be more separable in this space. The kernel is an N×N matrix, where N is the total number of samples. In KNWFE, a weight matrix is firstly defined, based on data.

$$\lambda_l^{(i,j)} = \frac{\left[K_{ll}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll} - 2(K^{(i,j)}W^{(i,j)^T})_{ll}\right]^{-1/2}}{\sum_{t=1}^{N_i}\left[K_{tt}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{tt} - 2(K^{(i,j)}W^{(i,j)^T})_{tt}\right]^{-1/2}} \quad (1)$$

Where l represents a datum, $i = 1,2,\dots,L$, $j = 1,2,\dots,L$, L is the number of classes, $K^{(i,j)}$ is a part of kernel matrix, K, and $W^{(i,j)}$ which is shown in (2). Matrix $\Lambda^{(i,j)}$, is then defined as in (3) which is used in the process [26].

$$W^{(i,j)} = \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{N_i1}^{(i,j)} & \cdots & w_{N_iN_j}^{(i,j)} \end{bmatrix} \quad (2)$$

$$\Lambda^{(i,j)} = diag\left\{\frac{\lambda_1^{(i,j)}}{N_i}, \dots, \frac{\lambda_{N_i}^{(i,j)}}{N_i}\right\} \quad (3)$$

Where $w_{lk}^{(i,j)}$ is defined in (4), and N and N_i are total number of data, and number of data in class i, respectively.

$$w_{lk}^{(i,j)} = \frac{(K_{ll}^{(i,i)} + K_{kk}^{(j,j)} - 2K_{lk}^{(i,j)})^{-1}}{\sum_{t=1}^{N_j}(K_{ll}^{(i,i)} + K_{tt}^{(j,j)} - 2K_{lt}^{(i,j)})^{-1}} \quad (4)$$

To obtain a transformation matrix, it is firstly needed to calculate the two matrices W and B (equations (5)-(12)).

$$W = W_1 - W_2 - W_2^T + W_3 \quad (5)$$

$$W_1 = diag\{P_1\Lambda^{(1,1)}, \dots, P_L\Lambda^{(L,L)}\} \quad (6)$$

$$W_2 = diag\{P_1\Lambda^{(1,1)}W^{(1,1)}, \dots, P_L\Lambda^{(L,L)}W^{(L,L)}\} \quad (7)$$

$$W_3 = diag\{P_1W^{(1,1)^T}\Lambda^{(1,1)}W^{(1,1)}, \dots, P_LW^{(L,L)^T}\Lambda^{(L,L)}W^{(L,L)}\} \quad (8)$$

$$B = B_1 - B_2 - B_2^T + B_3 \quad (9)$$

$$B_1 = diag\left\{P_1\sum_{j=1}^{L}\Lambda^{(1,j)}, \dots, P_L\sum_{j=1}^{L}\Lambda^{(L,j)}\right\} \quad (10)$$

$$B_2 = \begin{bmatrix} P_1\Lambda^{(1,1)}W^{(1,1)} & \cdots & P_1\Lambda^{(1,L)}W^{(1,L)} \\ \vdots & \ddots & \vdots \\ P_L\Lambda^{(L,1)}W^{(L,1)} & \cdots & P_L\Lambda^{(L,L)}W^{(L,L)} \end{bmatrix} \quad (11)$$

$$B_3 = \sum_{i=1}^{L} P_i diag\{W^{(i,1)^T}\Lambda^{(i,1)}W^{(i,1)}, \dots, W^{(i,L)^T}\Lambda^{(i,L)}W^{(i,L)}\} \quad (12)$$

In the above equations, P_i is the probability of i^{th} class. The following steps are taken to obtain the transformation matrix (A) [26].

If the transformation matrix is derived according to Fisher's relationship as follows (Equation (13)), then it is necessary to use the decomposition of the eigenvalue and eigenvector to obtain P and U:

$$A = PU \quad (13)$$

Where P is the eigenvector of the kernel matrix and U is the eigenvector of equation (15). These eigenvectors are arranged based on the highest eigenvalues, and the eigenvector whose eigenvalues are zero or close to zero is eliminated.

$$K = P\Gamma P^T \quad (14)$$

$$(\Gamma P^T(B - W)P\Gamma)U = \lambda(\Gamma P^T W P\Gamma)U \quad (15)$$

Then, the transformation equation will be:

$$y = A^T\begin{bmatrix}\kappa(x_1, z) \\ \vdots \\ \kappa(x_N, z)\end{bmatrix} \quad (16)$$

As a result of Equation (1), the KNWFE algorithm assigns greater weights to the samples close to the center of the class, in contrary to the boundary samples. Meanwhile, as it is illustrated in Fig. 1 the boundary samples are more determinant than samples close to the center of the class. Furthermore, this method does not differentiate between classes that are close to each other, and far apart in the production of the between-class scattering matrix. Whiles, it can have a significant effect on determining the final weight of the sample. The following is a description of the proposed method for solving the problems of this algorithm.

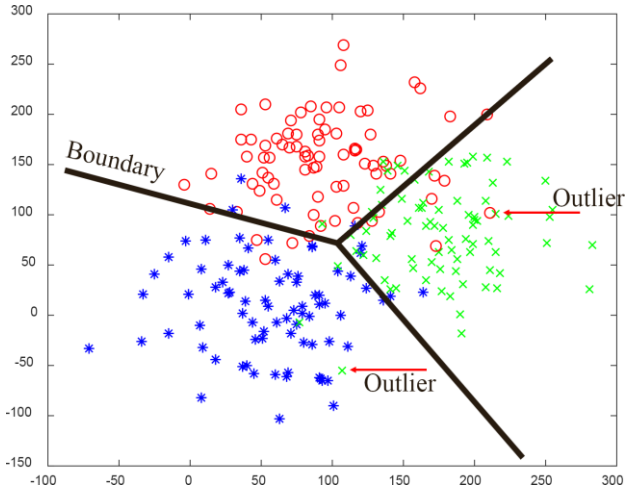

Fig. 1. The importance of boundary s in SVM classification

3- Optimized KNWFE

To solve the first problem of assigning more weight to data close to the center of the class, we use a function with Rayleigh distribution. In such a way, we pass the scattering matrix elements through this function and give higher weights to the samples near the class boundary. We then discard twenty percent of the samples in each class to prevent high weight assignment to outliers. The proposed formulations are also designed so that they take into account the distance of classes.

We define the weights $\gamma_l^{(i,j)}$ corresponding to $\lambda_l^{(i,j)}$ and another matrix $\gamma_l^{un\,(i,j)}$ so that $\gamma_l^{un\,(i,j)}$ are unnormalized weights of $\gamma_l^{(i,j)}$, and then we will have:

$$\gamma_l^{(i,j)} = \frac{\left[K_{ll}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll} - 2(K^{(i,j)}W^{(i,j)^T})_{ll}\right]^{-1/2}}{\sum_{t=1}^{N_i}\left[K_{tt}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{tt} - 2(K^{(i,j)}W^{(i,j)^T})_{tt}\right]^{-1/2}} \quad (17)$$

where i and j are indices of i^{th} and j^{th} classes, l is the index of the datum, and N_i is the total number of data in the class.

$$\gamma_l^{un\,(i,j)} = \left[K_{ll}^{(i,i)} + (W^{(i,j)}K^{(j,j)}W^{(i,j)^T})_{ll} - 2(K^{(i,j)}W^{(i,j)^T})_{ll}\right]^{-1/2} \quad (18)$$

First, it is needed to modify $\gamma_l^{(i,j)}$, which is the membership degree of each data in its class, as follows. In this case, the more the data is away from the center of the class (boundary data), it will gain more weight.

$$\gamma_l^{(i,i)} = \frac{\max\left(\gamma_l^{(i,i)}\right) - \gamma_l^{(i,i)}}{\sum_{t=1}^{N_i}\left\{\max\left(\gamma_t^{(i,i)}\right) - \gamma_t^{(i,i)}\right\}} \quad (19)$$

The following equation (similar to the Rayleigh distribution function) is used to weaken the effect of distorted and noisy data and remove them from the data set. In this case, we must apply this relation to the entire matrix of $\gamma_l^{(i,i)}$.

$$\gamma_l^{(i,j)} = \frac{\gamma_l^{(i,j)}}{0.606\sigma} e^{-\frac{\gamma_l^{(i,j)^2}}{2\sigma^2}} \quad (20)$$

The value of σ is the percentage of noisy data deletion. We chose this so that to consider 20% of the data of each class as offset data. That is, due to the above relationship, 20 percent of the data in a class will be weighed less and will be considered as noisy data and will not be known as boundary data. We apply the relation to the total weight of $\gamma_l^{un\,(i,j)}$. Dividing by 0.606, we normalized those weights:

$$\gamma_l^{un\,(i,j)} = \frac{\gamma_l^{un\,(i,j)}}{0.606} e^{-\frac{\gamma_l^{un\,(i,j)^2}}{2\sigma^2}} \quad (21)$$

The final weights of the data for the within-class and between-class values will be as follows:

$$\lambda_l^{(i,i)} = 0.5(\gamma_l^{(i,i)} + \max_{\forall\, i\neq j}\gamma_l^{(i,j)})\,(\max_{\forall\, i\neq j}\gamma_l^{un(i,j)})^r \quad (22)$$

$r \in [0,\infty)$

where the r parameter increases the effect of non-normalized weights on the total weights.

$$\lambda_l^{(i,j)} = 0.5(\gamma_l^{(i,i)} + \max_{\forall\, i\neq j}\gamma_l^{(i,j)})(\gamma_l^{un(i,j)})^r \quad (23)$$

$r \in [0,\infty)$

We then normalize equations (22) and (23) and obtain equations (24) and (25). Multiplying the two terms, we can change the weight between the terms, by raising one of the terms to the power of r. One may change the power r to increase the effect of $\gamma_l^{un\,(i,j)}$. Thus, the weights $\lambda_l^{(i,i)}$ and $\lambda_l^{(i,j)}$ are replaced in the original algorithm and improve the results. The next steps are similar to the original algorithm to obtain the conversion matrix.

$$\frac{\lambda_l^{(i,i)}}{\sum_l\sum_i\lambda_l^{(i,i)}} \to \lambda_l^{(i,i)} \quad (24)$$

$$\frac{\lambda_l^{(i,j)}}{\sum_l\sum_i\sum_{j\neq i}\lambda_l^{(i,j)}} \to \lambda_l^{(i,j)} \quad (25)$$

Superpixel segmentation algorithms were used along with the proposed kernel to increase the efficiency of the classification system [27]. This algorithm segments the HSI into a large number of superpixels. A superpixel consists of a combination of many contiguous pixels that have similar properties. Due to a large number of HSI bands, direct segmentation is not possible. Hence, we reduce the dimensionality using the proposed combined non-parametric kernel (CNPK) and classify it using SVM.

4- Experiments

We used three sets of HSI to evaluate the effectiveness of the proposed OKNWFE. The first set is taken from a forest-agricultural area in the northeast of the Indiana state, using the AVIRIS sensor in 1992. This image has a 220 band and 145×145 pixels. The dataset has 16 different classes and 10366 samples. Due to the absorption of radiations by the atmosphere, some of the bands are highly noisy and do not contain reliable information. Therefore, we reject the 30 noisy bands to improve the classification. The secondary data belongs to an area at Pavia University. This image is of size 610×340 and the high resolution of 1.3 meters per pixel in each band. The remaining number of channels after removing the noisy bands is 103, with a spectral range of 0.43 to 0.86 micrometers. This data includes nine different classes which are: Asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, brick blocks, and shadows [3]. The third data belongs to the Pavia urban area, which is a 115 dimensional and 1096×715 pixel image. Removing the noisy bands, 102 bands remained for the image. This data includes nine different classes, which are water, trees, asphalt, brick blocks, bitumen, tiles, shadows, meadows, and soils.

Table I. The best-tuned used for OKNWFE and KNWFE for the AVIRIS dataset

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	6.6	0.5
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	520000	500000
Number of k in k-NN	7	7
Distance type	Minkowski	Minkowski
Order	3	3

Table II. The best-tuned parameters for the optimized KNWFE and the KNWFE for the Pavia University Dataset

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	2000	220
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	10000	10000
Number of k in k-NN	7	7
Distance type	Minkowski	Minkowski
Order	3	3

4-1- Parameter Setting

As in [26], we used 8 of 16 classes in AVIRIS data to evaluate the baseline and the proposed method. The simulation time for the AVIRIS data and the KNWFE method was 43 minutes, and for the OKNWFE method it was 57 minutes; for the data of the Pavia University and the KNWFE 82 minutes; for the 108-minute OKNWFE method, for the Pavia urban area data and the KNWFE method, 104 minutes, for the method OKNWFE is 130 minutes. The experiments were carried out using Core i5 3210M CPU and 6GB of RAM under the plate of MATLAB. Selected classes are, Corn-no till, Corn-min till, grass, Hay-windrowed, Soybeans-notill, Soybeans-min till, Soybeans-clean till and Woods, which are labeled class1 through class8 respectively. One thousand samples were used for each class of PAVIA urban area data. For each of the three datasets of each class, 300 samples were used to obtain the transformation matrix using the algorithm, and 350 samples were used to learn the SVM classifier. Since training data are randomly selected to train the classifier, 5-fold cross-validation is used to improve predictive performance. The Gaussian kernel sigma value variations are set at {0.02, 0.2, 2, 20, ..., 2000} for the OKNWFE method and {0.02, 0.2, 2, ..., 220} for KNWFE. The range of variations of the Gaussian kernel sigma value of the SVM class for both OKNWFE and KNWFE methods is {0.2, 2, 20, ..., 200}. Also, the range of variations of C value for both OKNWFE and KNWFE is {0.1, 1, 10, 100, 10^3, …, 10^6}. The number of neighbors in the k-NN is also {1, 3, 5, 7, 9, 11}, and the range of Minkowski distance order changes is {2, 3, 4, 5}. We have randomly test values of the σ of the Gaussian kernel for both the proposed OKNWFE and KNWFE. In the classification experiments, the best empirical values of σ are used (Table I &Table II).

Table III. The best-tuned parameters for the optimized KNWFE and KNWFE for the Pavia urban area data set

Method	***OKNWFE***	***KNWFE***
Kernel type	Gaussian	Gaussian
Value of σ	1300	2
SVM kernel type	Gaussian	Gaussian
SVM kernel width	20	20
Value of C	10000	10000
Number of k in k-NN	7	9
Distance type	Minkowski	Minkowski
Order	3	3

Table IV. The results obtained from the optimized KNWFE simulation and KNWFE for AVIRIS data

Criterions	***Overall Accuracy (%)***		***Average accuracy (%)***		***Kappa coefficient***	
Classifier	SVM	k-NN	SVM	k-NN	SVM	k-NN
OKNWFE	87.04	82.15	90.36	86.58	0.8462	0.7889
KNWFE	79.49	76.08	81.57	81.36	0.7569	0.7184

Table V. The results obtained from the optimized KNWFE simulation and KNWFE for data from the University of Pavia

Criterions	***Overall Accuracy (%)***		***Average accuracy (%)***		***Kappa coefficient***	
Classifier	SVM	k-NN	SVM	k-NN	SVM	k-NN
OKNWFE	88.73	81.10	91.86	87.26	0.8542	0.7585
KNWFE	83.88	71.08	88.45	78.12	0.7923	0.6254

Table I shows the empirically determined parameters for KNWFE and the proposed OKNWFE methods for the AVIRIS dataset. Conducting experiments on the AVIRIS dataset, it is empirically determined that the optimum value of σ for the Gaussian KNWFE is 0.5, while it is 6.6 for the Gaussian OKNWFE, and the value of *C* is 500,000 and 520,000 for KNWFE and OKNWFE respectively. The rest of the parameters are the same for both methods.

Table VI. The classification accuracy of each class of AVIRIS data using the SVM for both methods (%)

Class	1	2	3	4
OKNWFE	83.12	88.12	95.97	100
KNWFE	71.96	72.06	72.63	98.97
Class	5	6	7	8
OKNWFE	90.59	78.11	93.15	93.81
KNWFE	89.56	71.88	81.59	93.89

Table VII. The classification accuracy of each class of AVIRIS data using the k-NN for both methods (%)

Class	1	2	3	4
OKNWFE	73.01	81.53	95.77	99.59
KNWFE	59.55	73.86	94.56	99.79
Class	5	6	7	8
OKNWFE	89.87	72.24	90.71	89.95
KNWFE	85.95	66.73	83.87	86.55

Table VIII. The confusion matrix of AVIRIS data classification using the OKNWFE and SVM classifier

Class	1	2	3	4	5	6	7	8
1	1192	14	3	1	71	97	55	1
2	16	735	0	0	6	32	45	0
3	1	0	447	2	1	1	2	13
4	0	0	0	489	0	0	0	0
5	21	6	5	0	877	39	20	0
6	181	122	12	5	123	1928	97	0
7	14	12	3	0	6	7	572	0
8	0	0	74	6	0	0	0	1214

Table IX. The confusion matrix of AVIRIS data classification using the KNWFE and SVM classifier

Class	1	2	3	4	5	6	7	8
1	1032	79	4	0	114	105	100	0
2	52	601	2	1	11	86	81	0
3	0	0	361	1	0	0	2	133
4	0	0	4	484	0	0	0	1
5	36	4	6	0	867	28	27	0
6	237	135	20	4	177	1774	116	5
7	36	29	6	0	16	24	501	2
8	0	0	79	0	0	0	0	1215

Table III depicts that we empirically choose the value of 1300 and 2 for σ of OKNWFE and KNWFE, respectively. We also determine the number of K in KNN as 6 for OKNWFE and 9 for KNWFE. Table VI show the simulation results of both KNWFE and OKNWFE on all the three datasets. Table VI and Table VII how the SVM and k-NN classification accuracy on the AVIRIS data.

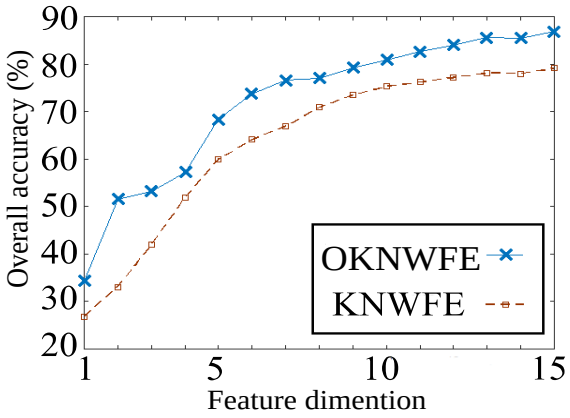

Fig. 2. Hugh diagram for OKNWFE and KNWFE using

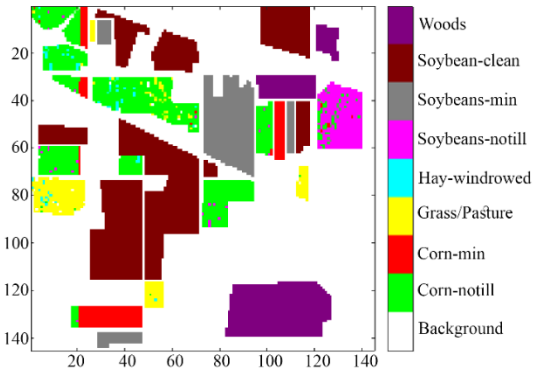

Fig. 3. OKNWFE classification map using SVM classification for the AVIRIS data

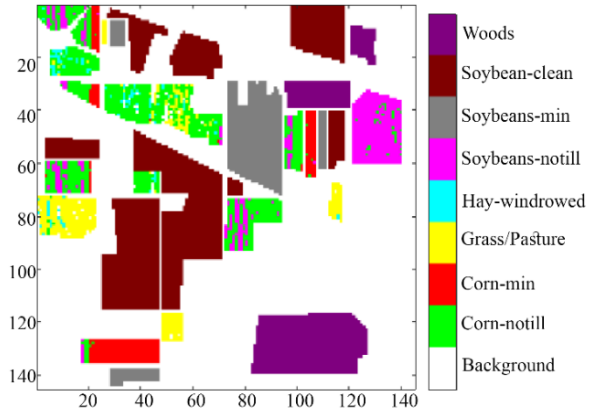

Fig. 4. Classification of the KNWFE method map using the SVM classification for the AVIRIS data

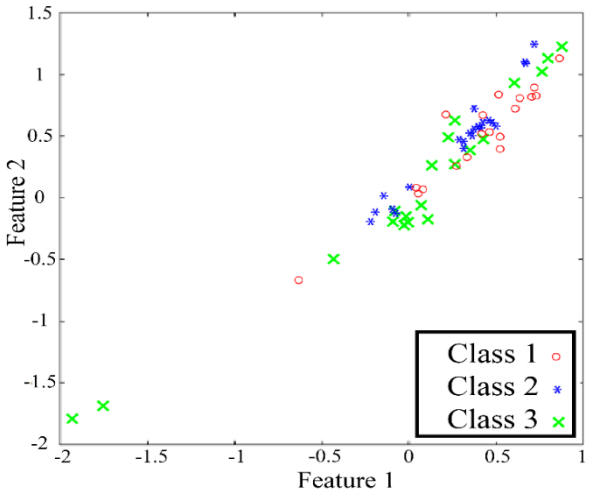

Fig. 6. Dispersion map of the KNWFE method for the AVIRIS data

4-2- Dispersion Map Analysis

To prevent the same results from being exceeded, we refrain from providing details of the results of other datasets. In the following, only the results of the SVM classification for the AVIRIS data are given. Table VIII and Table IX show the confusion matrices for both methods using the SVM classifier on the AVIRIS dataset. As the results of the experiments show, the proposed method with optimized data weights is superior to the KNWFE method. This improvement will come at the expense of increased computing.

The Hugh diagram is shown in Fig. 2. This figure shows the classification accuracy concerning the number of features. As it is clear from the curve, the accuracy of both the methods increases with the number of features until it goes to a so-called saturation. This also demonstrates that the OKNWFE method can achieve better performance for all the number of features.

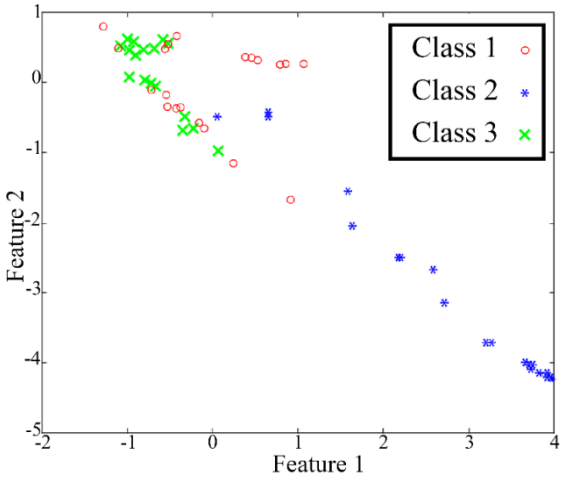

Fig. 5. Dispersion map of OKNWFE for the AVIRIS data

Fig. 3 and Fig. 4 represent the classification map, where each class represented with a color. Spots on some classes, which are more pronounced in the corn-notill class, indicate classification errors. As the comparison of the two images clearly shows, this error in OKNWFE is far less than in KNWFE. Fig. 5 and Fig. 6 show the dispersion map for both the methods using SVM classifier on AVIRIS data. As it is clear from Fig. 2, with the increase in the number of features, the separability of classes increases. As the number of features increases, the slope of the curve decreases and eventually reaches almost zero. It is also clear that the proposed method has more classification accuracy than the KNWFE method at each step and with the same number of attributes.

4-3- Experimental Results and Analysis

In this paper, we propose a hybrid non-parametric optimized kernel method for HSI classification, comparing its performance with some of the state-of-the-art methods of HSI classification as baseline methods. These methods include: SC-MK [28], RMK [29], RPNet [30]. Table X compares the performance of the base methods with the proposed CNPK method for the PAVIA university database. This table shows the classification efficiency of each class as well as the overall performance (OA). Each method was performed ten times using randomly selected samples to ensure the generality of the results, and we entered the average accuracy for each method in the table. The results show that the proposed method is more efficient in most classes than other methods. Also, in overall performance, the proposed method shows better results than all classes.

Table X Classification performance for PAVIA university dataset

Class name	***SC-MK***	***RMK***	***RPNet***	***CNPK***
Asphalt	0.8279	**0.9821**	0.952	0.9709
Meadows	0.9083	**0.9783**	0.9663	0.9668
Gravel	0.9176	0.9588	0.8856	**0.9781**
Trees	0.9652	0.905	0.9618	**0.9676**
Metal sheets	0.9999	0.9715	0.9634	**0.9999**
Bare soil	0.9711	0.9902	0.9088	**0.9937**
Bitumen	0.9601	0.9923	0.7825	**0.9994**
Bricks	0.9063	**0.9731**	0.9306	0.9725
Shadows	0.9682	0.5602	0.8222	**0.9927**
OA	0.9361	0.9235	0.9082	**0.9824**

The same experiment is performed on the PAVIA urban database. Table XI illustrates the classification results for this database.

Table XI Classification performance for PAVIA urban area dataset

Class name	***SC-MK***	***RMK***	***RPNet***	***CNPK***
Water	0.9992	0.9739	0.9952	**1.0000**
Tree	0.9186	0.8222	0.9008	**0.9573**
Asphalt	0.9723	0.9289	0.969	**0.9755**
Blocking Bricks	0.9904	0.9771	**0.9936**	0.9924
Bitumen	**0.9978**	0.9573	0.9768	0.9735
Tiles	**0.994**	0.9667	0.9618	0.9683
Shadows	0.9684	0.9752	0.9273	**0.9873**
Meadows	0.9831	0.9262	0.948	**0.99**
Bare Soil	0.9719	0.8162	0.9765	**0.9863**
OA	0.9773	0.9271	0.961	**0.9812**

5- Conclusion and Discussion

In this paper, we propose a feature extraction method that reduces the dimensions of a hyperspectral image so that the different segments of the image are better distinguishable. The method, which is called OKNWFE, results in the improvement of HSI classification. This improvement is obtained at the cost of an increase in computation complexity. As shown in Table IV to Table VI and Fig. 2 to Fig. 6, the OKNWFE method outperforms KNWFE. The dispersion map, drawn for the first and second characteristics, shows that the OKNWFE method provides better separation than the KNWFE method, and the classification map shows that the proposed method has less error in the classification of classes had. The experimental results suggest that the proposed method, in combination with the superpixel segmentation algorithms, has superior performance to the state-of-the-art systems for HIS classification. Based on the results and the proposed method, suggestions can be made for future research. As future work, one may use this kernel on CNNs for classifying hyperspectral images. This kernel can be an activation function on CNN. Adapting this kernel to achieve an optimal CNN could be the subject of future research. The type and number of other layers besides this activation function is another subject worth exploring. Calculating optimal kernel parameters for kernel-based methods using cross-validation is very time-consuming, so studying computation reduction methods can be a useful study. Besides, one may use synthetic kernels and find combination parameters using the classical optimization method.

References

[1] H. Li, H. Zhou, L. Pan, and Q. Du, "Gabor feature-based composite kernel method for hyperspectral image classification," vol. 54, no. 10, 2018, doi: 10.1049/el.2018.0272.

[2] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant Attribute Profiles: A Spatial-Frequency Joint Feature Extractor for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens., pp. 1–18, 2020, doi: 10.1109/TGRS.2019.2957251.

[3] S. Suresh and S. Lal, "A Metaheuristic Framework based Automated Spatial-Spectral Graph for Land Cover Classification from Multispectral and Hyperspectral Satellite Images," Infrared Phys. Technol., vol. 105, no. January, p. 103172, 2020, doi: 10.1016/j.infrared.2019.103172.

[4] P. Xiang et al., "Hyperspectral anomaly detection by local joint subspace process and support vector machine," Int. J. Remote Sens., vol. 41, no. 10, pp. 3798–3819, 2020.

[5] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," IEEE Geosci. Remote Sens. Mag., vol. 5, no. 1, pp. 8–32, 2017.

[6] L. Fang, Z. Liu, and W. Song, "Deep Hashing Neural Networks for Hyperspectral," IEEE Geosci. Remote Sens. Lett., vol. PP, pp. 1–5, 2019, doi: 10.1109/LGRS.2019.2899823.

[7] E. M. Paoletti, M. J. Haut, J. Plaza, and A. Plaza, "Deep&Dense Convolutional Neural Network for Hyperspectral Image Classification," Remote Sens., vol. 10, no. 9, pp. 1–21, 2018, doi: 10.3390/rs10091454.

[8] H. Lee, M. Kim, D. Jeong, S. Delwiche, K. Chao, and B.-K. Cho, "Detection of cracks on tomatoes using a hyperspectral near-infrared reflectance imaging system," Sensors, vol. 14, no. 10, pp. 18837–18850, 2014.

[9] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," IEEE Trans. Geosci. Remote Sens., vol. 42, no. 5, pp. 1096–1105, 2004.

[10] M. R. Almeida, L. P. L. Logrado, J. J. Zacca, D. N. Correa, and R. J. Poppi, "Raman hyperspectral imaging in conjunction with independent component analysis as a forensic tool for explosive analysis: The case of an ATM explosion," Talanta, vol. 174, pp. 628–632, 2017.

[11] Z. Chen, J. Jiang, X. Jiang, X. Fang, and Z. Cai, "Spectral-spatial feature extraction of hyperspectral images based on propagation filter," Sensors (Switzerland), vol. 18, no. 6, pp. 1–16, 2018, doi: 10.3390/s18061978.

[12] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: A Superpixelwise PCA Approach for

Unsupervised Feature Extraction of Hyperspectral Imagery," IEEE Trans. Geosci. Remote Sens., vol. 56, no. 8, pp. 4581–4593, Aug. 2018, doi: 10.1109/TGRS.2018.2828029.

[13] H. Su, S. Member, B. Zhao, Q. Du, P. Du, and S. Member, "With Local Correlation Features for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens., vol. PP, pp. 1–12, 2018, doi: 10.1109/TGRS.2018.2866190.

[14] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 6, pp. 1351–1362, Jun. 2005, doi: 10.1109/TGRS.2005.846154.

[15] M. Khodadadzadeh, P. Ghamisi, C. Contreras, and R. Gloaguen, "Subspace Multinomial Logistic Regression Ensemble for Classification of Hyperspectral Images," in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Jul. 2018, pp. 5740–5743, doi: 10.1109/IGARSS.2018.8519404.

[16] S. Song, H. Zhou, J. Zhou, K. Qian, K. Cheng, and Z. Zhang, "Hyperspectral anomaly detection based on anomalous component extraction framework," Infrared Phys. Technol., vol. 96, pp. 340–350, 2019, doi: 10.1016/j.infrared.2018.12.008.

[17] E. Blanzieri and F. Melgani, "Nearest Neighbor Classification of Remote Sensing Images With the Maximal Margin Principle," IEEE Trans. Geosci. Remote Sens., vol. 46, no. 6, pp. 1804–1811, Jun. 2008, doi: 10.1109/TGRS.2008.916090.

[18] D. Tuia and G. Camps-Valls, "Semisupervised Remote Sensing Image Classification With Cluster Kernels," IEEE Geosci. Remote Sens. Lett., vol. 6, no. 2, pp. 224–228, Apr. 2009, doi: 10.1109/LGRS.2008.2010275.

[19] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral Image Classification via.pdf," vol. 51, no. 1, pp. 217–231, 2013.

[20] X. Weng, W. Lei, and X. Ren, "Kernel sparse representation for hyperspectral unmixing based on high mutual coherence spectral library," Int. J. Remote Sens., vol. 41, no. 4, pp. 1286–1301, 2020, doi: 10.1080/01431161.2019.1666215.

[21] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal Patch Tensor Sparse Representation for Hyperspectral Image Super-Resolution," IEEE Trans. Image Process., vol. 28, no. 6, pp. 3034–3047, 2019, doi: 10.1109/TIP.2019.2893530.

[22] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens., vol. PP, pp. 1–11, 2018, doi: 10.1109/TGRS.2018.2841823.

[23] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "ISPRS Journal of Photogrammetry and Remote Sensing A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120–147, 2018, doi: 10.1016/j.isprsjprs.2017.11.021.

[24] B. Pan, Z. Shi, and X. Xu, "ISPRS Journal of Photogrammetry and Remote Sensing MugNet : Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, 2017, doi: 10.1016/j.isprsjprs.2017.11.003.

[25] O. Okwuashi and C. E. Ndehedehe, "Deep support vector machine for hyperspectral image classification," Pattern Recognit., vol. 103, pp. 2–25, 2020, doi: 10.1016/j.patcog.2020.107298.

[26] B. C. Kuo, C. H. Li, and J. M. Yang, "Kernel nonparametric weighted feature extraction for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 47, no. 4, pp. 1139–1155, 2009, doi: 10.1109/TGRS.2008.2008308.

[27] L. Sun, C. Ma, Y. Chen, H. J. Shim, Z. Wu, and B. Jeon, "Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 12, no. 6, pp. 1905–1919, 2019.

[28] T. Zhan, L. Sun, Y. Xu, G. Yang, Y. Zhang, and Z. Wu, "Hyperspectral classification via superpixel kernel learning-based low rank representation," Remote Sens., vol. 10, no. 10, p. 1639, 2018.

[29] J. Liu, Z. Wu, Z. Xiao, and J. Yang, "Region-based relaxed multiple kernel collaborative representation for hyperspectral image classification," IEEE Access, vol. 5, pp. 20921–20933, 2017.

[30] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," ISPRS J. Photogramm. Remote Sens., vol. 142, pp. 344–357, 2018.